

A档模型 UNet

--by 2213409 石彬辰

BUSI-256数据集

训练步骤

将数据集BUSI-256拷贝到项目根目录下,改变文件结构,images->images,masks->labels,新建子目录ImageSets/Segmentation

如下修改voc_annotation.py

voc_annotation.py

```
1  VOCdevkit_path      = 'BUSI-256' #'VOCdevkit'
2
3  if __name__ == "__main__":
4      random.seed(0)
5      segfilepath=os.path.join(VOCdevkit_path,'labels')
6      saveBasePath=os.path.join(VOCdevkit_path,'ImageSets/Segmentation')
```

由于部分labels中的遮罩图片为24位,修改voc_annotation.py如下部分,跳过这些图片
然后运行voc_annotation.py,生成对应的txt文件

voc_annotation.py

```
1  for i in tqdm(list):
2      name          = total_seg[i]
3      png_file_name  = os.path.join(segfilepath, name)
4      if not os.path.exists(png_file_name):
5          raise ValueError("未检测到标签图片%s, 请查看具体路径下文件是否存在以及后缀是否为png。"%(png_file_name))
6
7      png            = np.array(Image.open(png_file_name), np.uint8)
8      if len(np.shape(png)) > 2:
9          #print("标签图片%s的shape为%s, 不属于灰度图或者八位彩图, 请仔细检查数据集格式。"%(name, str(np.shape(png))))
10         #print("标签图片需要为灰度图或者八位彩图, 标签的每个像素点的值就是这个像素点所属的种类。"%(name, str(np.shape(png))))
11         continue
```

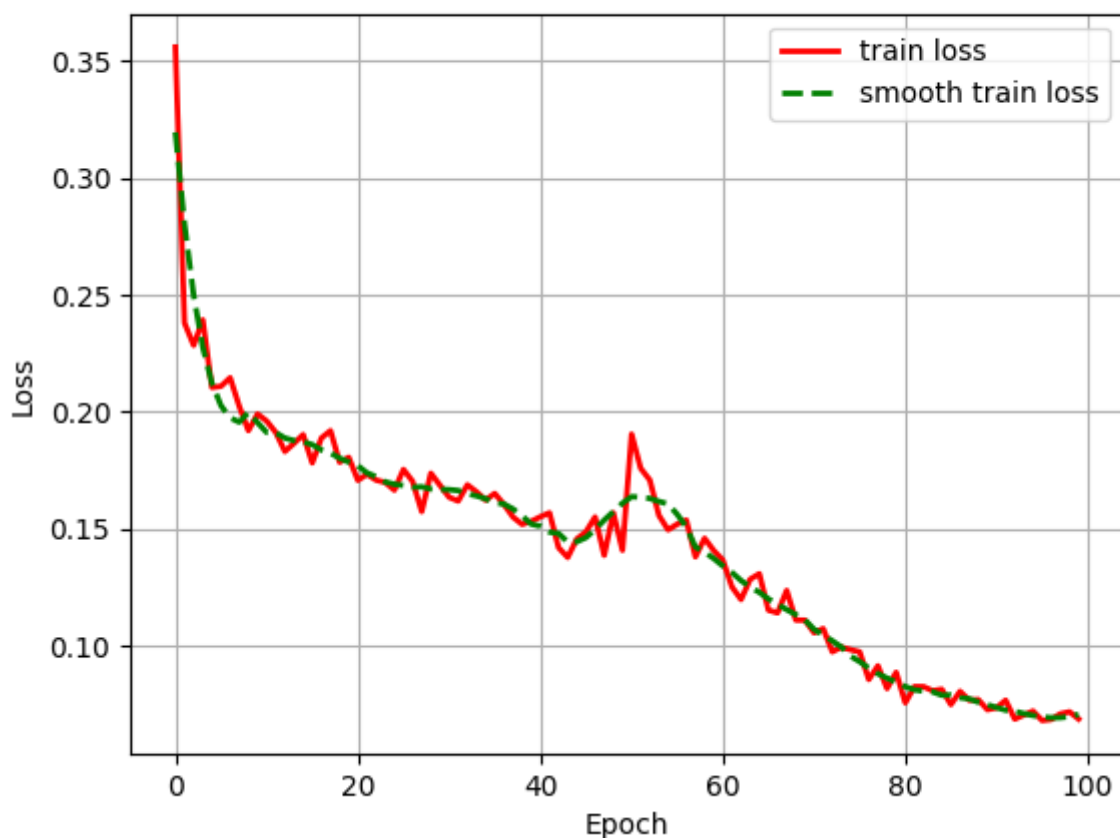
修改train_medical.py中的数据集路径,添加环境变量语句

train_medical.py

```
1  #-----#
2  #   数据集路径
3  #-----#
4  VOCdevkit_path = 'BUSI-256' #'Medical_Datasets'
5  os.environ["KMP_DUPLICATE_LIB_OK"]="TRUE"
```

运行train_medical.py开始训练,训练得到22个模型文件,best_epoch_weights.pth为所有模型中验证损失率最小的一个,last_epoch_weights.pth为最新的模型,其余二十个为训练100轮,每五轮保存一次的结果,将训练结果及日志另外保存到数据集目录下

从epoch_loss.png即训练轮次-损失率图像可以看出损失率随着训练轮次的变化,在40~60轮loss发生较大变化,在80轮后loss趋于平缓



epoch_loss.png

从训练日志可以看出epoch=96时验证损失率最小,最后一次取到最佳模型

代码块

```

1 Epoch 96/100: 0%|          | 0/283 [00:00<?, ?it/s<class 'dict'>]Start Train
2 Epoch 96/100: 100%|██████████| 283/283 [01:44<00:00, 2.71it/s,
  f_score=0.907, lr=1e-6, total_loss=0.068]
3 Epoch:96/100
4 Total Loss: 0.068
5 Save best model to best_epoch_weights.pth

```

预测步骤

新建predict_all.py用于预测所有图像

```

predict_all.py

1  import shutil
2
3  from PIL import Image
4  import os
5
6  from unet import Unet
7  if __name__=='__main__':
8      mode = "predict_all"
9      #image_dir='isic2018/test/images/'
10     #save_dir='isic2018/test/images_predict/'
11     image_dir='BUSI-256/images/'
12     save_dir='BUSI-256/images_predict/'
13     if os.path.exists(save_dir):
14         shutil.rmtree(save_dir)
15     os.mkdir(save_dir)
16     unet=Unet()
17     dir_list=os.listdir(image_dir)
18     for img in dir_list:
19         try:
20             image = Image.open(image_dir+img)
21         except:
22             print('Open Error! Try again!')
23             continue
24         else:
25             r_image = unet.detect_image(image)
26             r_image.save(save_dir+img)

```

修改unet.py文件,将model_path修改为训练好的模型存放位置,num_classes修改为train_medical.py中num_classes的数值input_shape修改为输入图片的大小

```

unet.py

1  #-----#

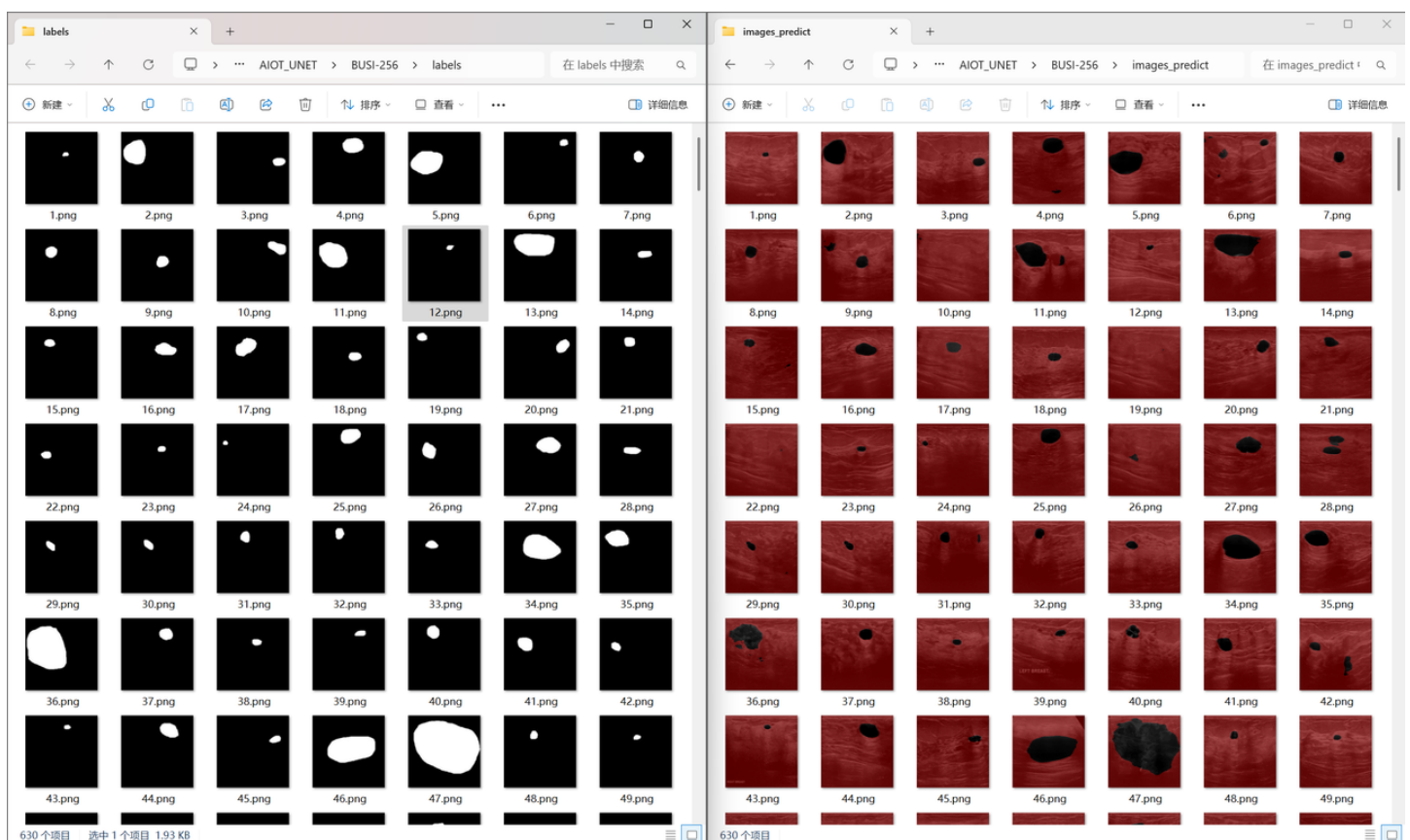
```

```

2  # model_path指向logs文件夹下的权值文件
3  # 训练好后logs文件夹下存在多个权值文件，选择验证集损失较低的即可。
4  # 验证集损失较低不代表miou较高，仅代表该权值在验证集上泛化性能较好。
5  #-----#
6  ##### "model_path"      : 'model_data/unet_vgg_voc.pth',
7  "model_path"      : 'BUSI-256/BUSI-256_train/best_epoch_weights.pth',
8  #-----#
9  # 所需要区分的类的个数+1
10 #-----#
11 "num_classes"      : 2,
12 #-----#
13 # 所使用的主干网络: vgg、resnet50
14 #-----#
15 "backbone"         : "vgg",
16 #-----#
17 # 输入图片的大小
18 #-----#
19 "input_shape"      : [256, 256],

```

运行predict_all.py,由于没有测试集,使用训练集进行预测,得到image_predict目录,对比mask和predict图像



左为mask,右为predict

isic2018数据集

训练步骤

将数据集isic2018/train拷贝到项目根目录下,改变文件结构,images->images,masks->labels,新建子目录ImageSets/Segmentation

如下修改并运行voc_annotation.py,生成对应的txt文件

```
voc_annotation.py

1  VOCdevkit_path      = 'isic2018/train' #'BUSI-256' #'VOCdevkit'
2
3  if __name__ == "__main__":
4      random.seed(0)
5      segfilepath=os.path.join(VOCdevkit_path,'labels')
6      saveBasePath=os.path.join(VOCdevkit_path,'ImageSets/Segmentation')
```

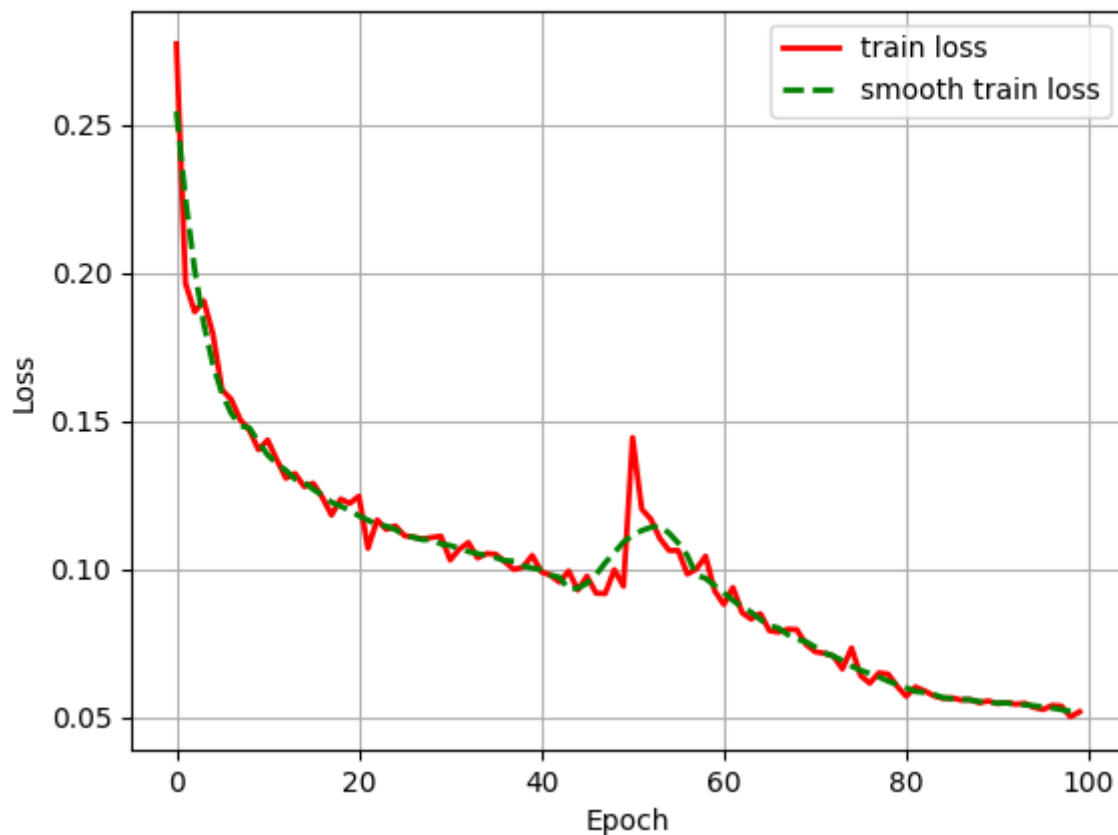
修改train_medical.py中的数据集路径

```
train_medical.py

1  #-----#
2  #   数据集路径
3  #-----#
4  VOCdevkit_path  = 'isic2018/train' #'BUSI-256' #'Medical_Datasets'
```

运行train_medical.py,训练得到22个模型文件,best_epoch_weghts.pth为所有模型中验证损失率最小的一个,last_epoch_weights.pth为最新的模型,其余二十个为训练100轮,每五轮保存一次的结果,将训练结果及日志另外保存到数据集目录下

从epoch_loss.png即训练轮次-损失率图像可以看出损失率随着训练轮次的变化,在40~60轮loss发生较大变化,在80轮后loss趋于平缓



epoch_loss.png

从训练日志可以看出epoch=99时验证损失率最小,最后一次取到最佳模型

代码块

```
1 Start Train
2 Epoch 99/100: 100%|██████████| 943/943 [01:46<00:00, 8.85it/s,
  f_score=0.956, lr=1e-6, total_loss=0.0502]
3 Epoch:99/100
4 Total Loss: 0.050
5 Save best model to best_epoch_weights.pth
6 Epoch 100/100: 0%|          | 0/943 [00:00<?, ?it/s<class 'dict'>]Start Train
7 Epoch 100/100: 100%|██████████| 943/943 [01:46<00:00, 8.85it/s,
  f_score=0.955, lr=1e-6, total_loss=0.0518]
8 Epoch:100/100
9 Total Loss: 0.052
```

预测步骤

由于isic2018数据集有测试集,本次预测使用测试集数据,将isic2018/test拷贝到项目中的isic2018目录下

修改unet.py

```

1 net.py -----#
2 # model_path指向logs文件夹下的权值文件
3 # 训练好后logs文件夹下存在多个权值文件，选择验证集损失较低的即可。
4 # 验证集损失较低不代表miou较高，仅代表该权值在验证集上泛化性能较好。
5 # -----#
6 ##### "model_path" : 'model_data/unet_vgg_voc.pth',
7 # "model_path" : 'BUSI-256/BUSI-256_train/best_epoch_weights.pth',
8 "model_path" : 'isic2018/isic2018_train/best_epoch_weights.pth',

```

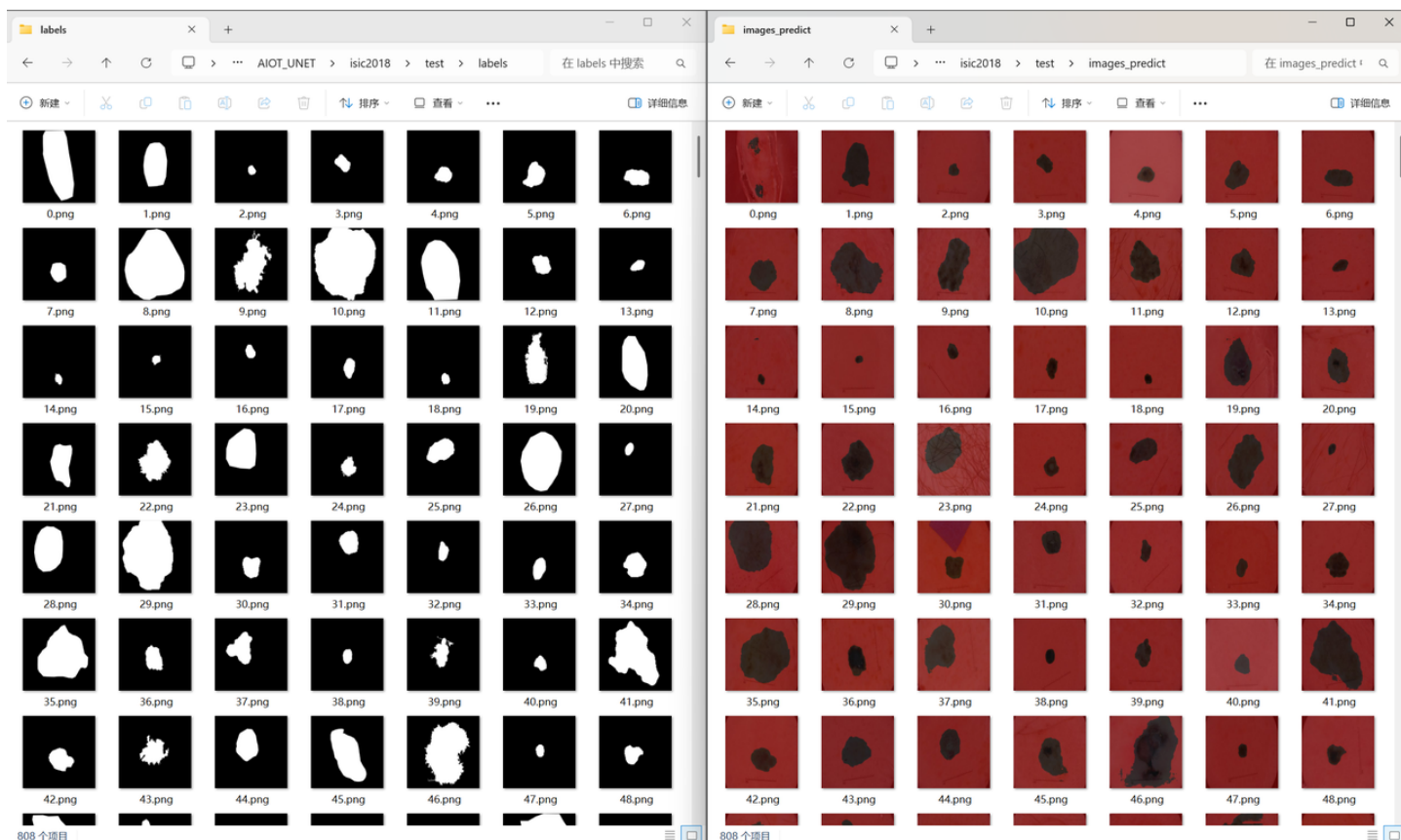
修改predict_all.py

```

predict_all.py
1 mode = "predict_all"
2 #image_dir='BUSI-256/images/'
3 #save_dir='BUSI-256/images_predict/'
4 image_dir='isic2018/test/images/'
5 save_dir='isic2018/test/images_predict/'

```

运行predict_all.py,使用测试集预测,获得预测图像,对比mask和predict图像



左为mask,右为predict

备注

- 医学图像不可分类,故无法进行计算miou等性能指标,只能用于观看数据集的训练效果
- BUSI-256只有训练集,故预测借用训练集,isic2018 train训练集用于训练模型,test测试集用于预测,预测图像保存在isic2018/test/images_predict目录下
- 两个数据集训练出来的模型及数据保存在各自子目录' 目录名+train' 下,训练过程信息保存在该子目录下的log.txt文件中
- 更新了需求文档requirement.txt